**Array Practice Problems**

1. Write a program in the following steps
2. Generates 10 Random 3 Digit number.
3. Store these random numbers into an array.
4. Then find the 2nd largest and the 2nd smallest element without sorting the array.

#!/bin/bash -x

for (( i=0;i<=10;i++ ))

do

randomNumber[$i]=$((RANDOM%900 + 100));

done

echo ${randomNumber[@]};

secondLargest=$(printf '%s\n' "${randomNumber[@]}" | sort -n | tail -2 | head -1);

secondSmallest=$(printf '%s\n' "${randomNumber[@]}" | sort -n | head -2 | tail -1);

echo "Second Largest: "$secondLargest;

echo "Second Smallest: "$secondSmallest;

1. Extend the above program to sort the array and then find the 2nd largest and the 2nd smallest element.

#!/bin/bash -x

for (( i=0;i<10;i++ ))

do

randomNumber[$i]=$((RANDOM%900 + 100));

done

echo ${randomNumber[@]};

randomNumberLength=${#randomNumber[@]};

for (( i=0;i<$randomNumberLength;i++ ))

do

for (( j=i+1;j<$randomNumberLength;j++ ))

do

if [ ${randomNumber[i]} -gt ${randomNumber[j]} ];

then

temp=${randomNumber[i]};

randomNumber[$i]=${randomNumber[j]};

randomNumber[$j]=$temp;

fi

done

done

echo "Array element after sorting are:" ${randomNumber[@]};

echo "Second largest element is:" ${randomNumber[$(($randomNumberLength-2))]};

echo "Second smallest element is:" ${randomNumber[1]};

1. Extend the Prime Factorization Program to store all the Prime Factors fo a number n into an array and finally display the output.

#!/bin/bash -x

read -p "Enter Number:" number;

k=0;

for (( i=2;i\*i<=$number;i++ ))

do

if [ $(($number%$i)) -eq 0 ]

then

factor=$i

count=0;

for(( j=1;j<=$factor;j++ ))

do

n=$(($factor%$j));

if [ $n -eq 0 ];

then

count=$(($count+1));

fi

done

if [ $count -eq 2 ];

then

primeFactors[$k]="$factor";

k=$(($k+1));

fi

fi

done

echo "Prime Factors of Number:" ${primeFactors[@]};

1. Write a Program to show the Sum of three Integer adds to ZERO.

#!/bin/bash -x

function findSumZero() {

for (( i=0;i<$(($arrayLength-2));i++ ))

do

for (( j=$(($i+1));j<$(($arrayLength-1));j++ ))

do

for (( k=$(($j+1));k<$arrayLength;k++ ))

do

if [ $(( ${array[$i]} + ${array[$j]} + ${array[$k]} )) -eq 0 ]

then

echo ${array[$i]} ${array[$j]} ${array[$k]}

fi

done

done

done

}

array[0]="0";

array[1]="-1";

array[2]="-3";

array[3]="2";

array[4]="1";

echo ${array[@]};

arrayLength=${#array[@]};

findSumZero ${array[@]} $arrayLength;

1. Take a range from 0 - 100, find the digits that are repeated twice like 33, 77, etc and store them in an array.

#!/bin/bash -x

counter=0;

index=0;

while [[ counter -le 100 ]]

do

(( counter++ ));

unitPlace=$(( $counter%10 ));

tenPlace=$(( $counter/10 ));

if [[ $unitPlace -eq $tenPlace ]]

then

twiceDigitArray[index]=$counter;

(( index++ ));

fi

done

echo "Digits that are repeated twice:" ${twiceDigitArray[@]};